A Framework for Congestion Control in ATM Networks

by

Uzzal Baruah

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

FEBRUARY, 1995

A Framework for Congestion Control in ATM Networks

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Master of Technology

by

Uzzal Baruah

under the guidance of

Dr. Dheeraj Sanghi

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

March 1995

Certificate

It is certified that the work contained in this thesis entitled **A Framework for Congestion Control in ATM Networks**, by **Uzzal Baruah (Roll No: 9311132)** under my supervision, and that this work has not been submitted elsewhere for a degree.

Dheeraj Sanghi

Dr. Dheeraj Sanghi
Assistant Professor,
Department of Computer Science and Engineering
IIT, Kanpur

March 1995

Abstract

Retransmissions of packets that are dropped by the network is the primary cause of decreased network throughput. To increase network throughput every attempt should be made to reduce and perhaps eliminate retransmission. Buffer overflow at the nodes cause cells to be dropped in the network. It is often seen that excepting the congested link buffer, other link buffers on the path of the virtual connections passing through the congested link are seldom used. In a properly designed network, the traffic load due to congestion should be distributed among all the buffers involving virtual connections. From these ideas we develop a congestion control scheme, Hop by Hop BECN (HHBECN), for loss sensitive traffic. A reactive control, HHBECN, forward the congestion notification by a congested switch towards the source by only one hop. This scheme ensures no cell loss for loss sensitive traffic, ensure high utilization of link, and capable of controlling transient and long term congestion unlike other proposed reactive controls.

Key-words and phrases : *ATM, Congestion Control, Resource Management*

Acknowledgements

I am extremely grateful to my supervisor, *Dr. Dheeraj Sanghi*, for introducing me to an exciting area like ATM. I would also like to thank him for continuous encouragement, resourceful discussions, and guidance to tackle the problems.

I am indebted to *Dr. D Manjunath* for participating in a series of long discussions, for shaping the scheme that I proposed in this thesis. He always encouraged my original thoughts and tried to bring out the best out of me. He was always available to discuss about thesis as well as other matters.

Thanks are due to all my friends and classmates who made my brief stay at IIT Kanpur a memorable one. I can not think of the days in IIT Kanpur without *Raja Shekhar.* I am very thankful to him for his help in various academic tasks, as well as moral support throughout these days. I extend my thanks to Raja Shekhar's family for creating a homely environment, for me, in IIT Kanpur.

I thanks *Singhai* for helping me to prepare this thesis report.

Finally, I thank my parents, and brother and sister for their continued cooperation during these days apart.

Feb 10, 1995 **Uzzal Baruah**

To my Parents
Brother and Sister

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Broadband Integrated Digital Service (B-ISDN) supports a variety of services with conflicting requirements. This large span of requirements demands a universal network, flexible enough to provide all the services in a uniform way. Two other parameters influencing the direction taken by the B-ISDN are, the fast evolution of semi-conductor and optical technology, and the evolution in the system concept ideas, e. g., the shift of superfluous transport functions to the edge of the network.

The need for a flexible network, the progress in technology and system concepts, led to the definition of Asynchronous Transfer Mode (ATM). The ATM concept is now accepted as the ultimate solution for the ISDN by CCITT[1].

ATM is primarily a switching technology, based on fixed size cells. Each cell is of size 53 bytes with 5 bytes header. All the cells arriving at a specific port, belonging to the same connection, are uniquely identified by two fields in the header — Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI). The switch maps the the incoming VPI and VCI fields with the help of a lookup table and forwards the cells to the outgoing port [1]. It should be noted that in ATM, the fixed small size cell helps in overcoming the problem of uncertain delay in the nodes.

B-ISDN networks, on top of ATM, are expected to provide the information transport to a rich mixture of services and applications. ATM provides a flexible means for supporting

[1]The International Consultive Committee for Telecommunications and Telegraphy

a continuum of transport rates, and also provides a potential efficiency from the statistical sharing of network resources, e. g., bandwidth and buffers. B-ISDN/ATM network has to be engineered to *fully exploit* this potential for efficiency to be competitive with the specialized private network alternatives. It is evident that the B-ISDN's goals of supporting diverse services and traffic mixes with an efficient network resource engineering, require an effective congestion control scheme to be designed.

1.1 Problem of Congestion Control in ATM

ATM allows statistical multiplexing of connections to take advantage of the variable information transfer rates generated by many applications. Many connections may implicitly share resources on the assumption that each connection requires these resources only for a small fraction of time. Statistical multiplexing has the potential for substantial gains in bandwidth. However, this results in a non-zero probability of cell-level overload or congestion. It is necessary to engineer the probability of congestion so that the impact of congestion on the end users is minimized. The congestion can be controlled to a limited extent by the provision of buffers. But buffering introduces additional delays and it cannot totally eliminate congestion, as there is still a non-zero probability of buffer overflow.

ATM shares much in common with conventional packet switching at lower rates, for which the subject of congestion control has been extensively studied. Most of the techniques applied to congestion control in conventional packet switching, do not work well for ATM networks. Some of the factors that render congestion control difficult in an ATM environment are as follows.

1.1.1 High Link Speeds

CCITT has recommended two speeds for B-ISDN accesses, one approximately at 155 Mb/s and the second at approximately 600 Mb/s. Internally, the links may operate in the Gb/s range. It is apparent that cell processing schemes in ATM must work at speeds comparable to the high link speeds. Thus the schemes need to be sufficiently simple to avoid the excessive processing time.

Another problem caused by the high link rate is increased propagation delay-bandwidth product, i. e., the amount of traffic that can be in transit during a propagation delay time. This limits the applicability of congestion control schemes like choke packet based scheme.

1.1.2 Multiple Service Requirements

B-ISDN is targeted to support a rich mixture of services and applications. Some services, such as voice and real-time video, have strict delay requirements. On the other hand, the asynchronous data traffic, for example, file transfer, has a stringent loss requirement. In managing a wide range of performance requirements, the main challenge is an equitable and efficient allocation of network resources to different services. A congestion control scheme for ATM networks must ensure that the performance requirements of a connection are satisfied independent of the type and number of the other multiplexed connections.

1.1.3 Diverse Traffic Characteristics

The traffic patterns generated by various applications are likely to vary from constant bit rate to extremely bursty. Moreover, these traffic streams are difficult to characterize because sufficient information is not known about their statistical behavior. An ATM congestion control scheme must make judicious choices of connection acceptance, multiplexing and various resource allocations to ensure that the diverse and unpredictable traffic characteristics are not detrimental to the network performance.

1.2 ATM Services and QoS Requirements

ATM defines four classes of traffic [1, 5] depending on various characteristics. The following description identifies the Quality of Service(QoS) requirements of different classes in terms of jitter and cell loss probability.

Class A : It is connection oriented Constant Bit Rate (CBR) traffic, having strong timing relation between source and destination. Constant bit rate video, PCM encoded voice traffic and emulation of T carrier public network circuits are some examples of Class

A traffic. This class of traffic has bounded delay and strict cell loss requirements.

Class B : This class corresponds to connection oriented Variable Bit Rate (VBR), where a strong timing relation is required. Example of such traffic includes variable bit rate voice and variable bit rate video. This class traffic has also bounded jitter and strict cell loss requirements.

Class C : This class is also connection oriented VBR, but there is no timing relationship between source and destination. TCP/IP, X.25, and Frame Relay are the examples of this class. This class of traffic can tolerate higher delay. Though this class of traffic is "loss sensitive", higher layer can take care of the loss through retransmission.

Class D : This is connectionless VBR data traffic without any timing relationship between source and destination. Connectionless LAN traffic is the example of this class of traffic. The QoS requirement of this class of traffic is same as the Class C.

From QoS requirement point view, we regiouped these four classes into 3 classes: class 1 corresponds to class A traffic, class 2 to class B, and class C and class D grouped into class 3.

1.3 Overview of Resource Management

This section gives a brief description of various steps involved in resource management in ATM. In a typical ATM network, the source sends out a request to the destination, specifying the QoS requirements in a prespecified traffic descriptor[2]. All the network elements along the path process the request sent by the source and determine whether to accept the request depending on the availability of resources. The network elements must ensure that there will be no degradation in the QoS of existing connections if the new connection is accepted. Once the connection is accepted, the network must monitor the sources for a possible violation of the contract made at the time of connection establishment. This is called *policing*. Policing plays an important role to guarantee QoS for the virtual connections, because violation of contract may degrade the QoS promised for other connections.

[2]It gives a description of the source, which may include the peak rate and average rate of cell generation, burst size and inter burst gap, bounds on jitter and cell loss and other similar information.

ATM supports diverse classes of services. Therefore, each class should be given different priority for discarding cells in case of buffer overflow to maintain the QoS of each class. Also different priorities must be given to schedule the cells for transmission to provide requested QoS for different classes of traffic.

All the above mentioned schemes are preventive resource management scheme. The other type of resource management, the reactive control, takes action after congestion is detected. It was felt that reactive control alone would fail to control congestion as ATM provides real time services also. However, it was found that reactive control schemes can improve network performance in case of sustained congestion. A more detailed description of reactive control schemes, coupled with preventive control methods, and their importance in ATM networks is given in the next chapter.

1.4 Summary

If there is only delay sensitive, variable rate traffic in the network, it is not possible to increase the link utilization more than a certain level. The peak rate of this kind of traffic may be as high as 20 times the average rate. Apparently, such cases would give very low statistical gain irrespective of the scheduling scheme used. With the introduction of loss sensitive, higher delay traffic, buffer link utilization can be increased by using large but finite buffers. Unfortunately, none of the existing congestion control schemes guarantees no cell loss for loss sensitive traffic. As a result, higher layers retransmit the lost cells. In fact, the loss of one cell may result in retransmission of a large number of cells. For the transfer of large files, the retransmission places significant load on the network [26]. Obviously, retransmission is an overhead which should be minimized and possibly, eliminated.

Our interest in this thesis is to use some form of simple admission control policies and use a variation of the basic explicit congestion notification to provide loss free service to loss-sensitive traffic like data, while not affecting the QoS of the existing and keeping the link utilization high. The scheme suggested in this thesis for loss sensitive traffic is as follows: In case of congestion in an outgoing link, the switch notifies the congestion to the previous switches of the connections going through the congested link, rather than these

sources of the connections. The previous switches will forward this congestion notification one hop towards the sources depending on their own congestion status. In this way, we eliminate the delay-bandwidth product from congestion switch to source to delay-bandwidth product between two consecutive switches, for an action to take effect taken due to congestion. Therefore, a switch must have enough buffers to store as many cells as can be in transit between the switch and the previous switches, provided the previous switches stops transmitting as soon as they receive the congestion notification. Hence, for every link, a switch requires a specific number of buffers to guarantee no loss for loss sensitive traffic. This scheme ensures no loss for loss sensitive traffic and also guarantees faster delivery of data. The proposed methodology increases the hardware complexity of the switches. The memory requirement in a switch is of the order of the number of virtual connections passing through the switch, while the order of buffer requirement is about six times the sum of the peak rates of the virtual connections for each link.

The related work done in the area of congestion control in ATM networks is reported in Chapter 2. In Chapter 3 we describe our proposed scheme and a framework for congestion control using it. A simulator for ATM networks, AtmSim, has been developed at I. I. T., Kanpur. In Chapter 4, we discuss the modifications and extensions made to AtmSim to incorporate the above mentioned scheme. In Chapter 5 we discuss the results of our experiment and compare it with some other schemes that have been described in literature. The conclusions and scope for further work are presented in Chapter 6.

CHAPTER 2

Related Work

The congestion control functions are classified into two categories: preventive and reactive. The preventive controls take actions to prevent congestion from occurring while reactive controls attempt to recover from congestion once it has taken place. There are two main levels of congestion control in ATM networks as given below.

- **Call-Level Control** : The long term congestion can be avoided through preventive control by regulating the admission of new calls into the network.
- **Cell-level Control** : Congestion control at the cell-level deals with short term congestions. This can be preventive or reactive.

For each call there should be a notion of service contact, specifying a set of service parameters agreed upon between the network and the user. The service parameters are chosen during the call establishment procedure, and renegotiation of these parameters may be repeated during the call. The service parameters given in terms of well chosen descriptor, allows as unambiguous agreement as possible between the network and user. This facilitates limiting the traffic which the network is expected to carry. The choice of traffic descriptor should balance the requirement enforceability, unambiguity, ability of the end users to specify it and usability by the network for making acceptance/denial decisions. The network should be able to identify the traffic in excess of the service contract and to prevent the excessive traffic from adversely impacting the transport of other traffic. More detailed discussion on traffic monitoring is given in the later sections. The performance

parameters specified by the user include the throughput requirements, delay, jitter, and loss tolerance.

The performance parameters help in potential network call-level controls such as admittance or denial of a call, or negotiation for a set of service parameters that the network can support. Furthermore, the service needs of a call, or the services, the network is willing to commit, may change during a call's lifetime. Thus, there the within-call parameter may be potentially renegotiated, allowing the network to satisfy the dynamically changing service needs of call.

2.1 Connection Admission Control

In preventive type of control, traffics are prevented from entering the network at the call level control. During the connection establishment phase, the connection are admitted in such a way that probability of congestion occurring is acceptable. This is known as Connection Admission Control (CAC). There are various connection admission control schemes being proposed in literature. New calls are admitted as long as resources are available. If the calls are admitted using equivalent bandwidth (estimated bandwidth that is expected to require), a new call is admitted if the equivalent bandwidth of the new call does not result in the expected bandwidth usage for a link exceeding the threshold of available bandwidth. This scheme is simple but may not fairly admit the calls. Accepting a smaller number of calls requiring a large amount of bandwidth can block several smaller calls. This makes call blocking probability unacceptably high. Moreover, a disappropriate amount of bandwidth at a particular node may go to a small set of origin-destination pairs and other origin-destination pairs may not have enough bandwidth. Two schemes that deal with each of these problems are described below.

To address the first problem, it was proposed that a call should be admitted if its bandwidth requirement doesn't exceed some percentage of available bandwidth [14]. The author showed, through simulation that the arrival rate of high bandwidth call does not affect the blocking probability of lower bandwidth calls under this scheme. However, this at the expense of higher blocking probability for high bandwidth calls.

The *Virtual Path (VP)* allocation scheme address the second problem. VP is a logical link between a source and destination, established on a long term basis and consisting of a number of calls. Each VP can be allocated some small amount of bandwidth [24]. The remaining bandwidth can be allocated to VPs upon demand as their bandwidth requirement exceeds initial allocations. This dynamically allocated bandwidth is released as soon as the VP no longer requires it.

The disadvantages of the CAC schemes described above and other admission control schemes are the amount of memory required and added processing overhead. The process of admission control is generally performed at each node along a proposed route whether it uses bandwidth allocation or some measure of expected delay and loss. Each node must keep an account of the resources it has allocated and process call-setup packets. If a call is rejected, then a message must be sent back, through the same route, so that intermediate node can free the resources. Each node must also process call tear-downs.

An alternative scheme [20] replaces node-by-node admission control by a pseudopackets process at the source and destination. Intermediate node do not have to perform costly operations on pseudopackets, called scouts, with similar traffic characteristics as the desired connection (such as average bit rate and burstiness), is transmitted on the desired route. If the scout packet finds congestion along the path, it will either return to its source after it exceed specified maximum delay or be dropped at the congested link(s). In either case, the call is rejected. If no congestion is detected, then the call is admitted.

The scout packet method of call admission and set-up is interesting but unproven. Its statistical performance is unknown and it can falsely admit a call if a large number of calls are idle when the scout packet are sent. Alternatively, it can falsely reject a call if there was a temporary congestion in the network caused by a large number of active calls, even when there is generally enough capacity in the network to accommodate a new call.

Call-level control is a complex issue. A trade off must be made in performance and overhead in both admission control and information passing. The admission control scheme may possibly need to account for any traffic shaping. In this case, the scheme for admission control needs to be made in conjunction with the smoothing scheme used.

2.2 Usage Parameter Control (UPC)

UPC represents the set of actions taken by the network to monitor and control traffic on an ATM connection in terms of cell traffic volume and cell routing validity. This function is also called *police function.* The main purpose of policing is to enforce the compliance of every ATM connection to its negotiated traffic contract. Without UPC, the QoS of already established connection could be seriously affected. This could occur due to a terminal equipment failure, excessive cell delay variation or even traffic abuse could seriously affect the QoS to other already established connection [9].

There has been a considerable discussion regarding which parameters should be controlled, peak rate, average rate burst length, etc. With potentially many thousand connections wanting to access to the network, a simple, fair, and effective algorithm is required.

There is a general agreement that the peak rate must be monitored for all the connections. This peak rate is determined by several factors such as physical access bandwidth, the function performed by the higher layer protocols, and the application itself. Therefore, this rate serves as a maximum possible transfer rate for the connection. While monitoring peak cell transfer rate, a certain amount of tolerance must be included to account for cell delay variation and jitter. The network must take some operational steps for those connections which violates exceed the connection traffic descriptor.

While traffic policing emphasizes bandwidth enforcement, traffic shaping emphasizes reducing the burstiness and improving the throughput. This can be achieved by using buffering to transmit some constant number of bits in a fixed interval. Traffic shaping introduces delay. Therefore, traffic shaping is applicable to only to delay insensitive traffic.

2.2.1 Leaky Bucket Scheme

The leaky bucket scheme is one way to ensure that a source does not exceed allocated parameters. In a leaky bucket scheme, a cell is accepted only when it can draw a token from a token pool, called the leaky bucket. The tokens are generated at the average data transfer rate r agree upon at the time of admission in the network and are stored in the

token pool. The pool has a finite size denoted by δ. After filling the token pool, additional tokens are discarded. δ can be seen as the maximum allowable burst length(neglecting the tokens admitted during the burst) since a maximum of δ cells may be transmitted at one time. A token pool can be implemented using a counter that is increased when the tokens are generated and is decreased when the tokens are used.

Policing can be combined with shaping traffic in a system in which cells are queued instead of being discarded when the token pool is empty [3]. The cell blocking probability, the probability that a cell arrives at an empty token pool, depends on the capacity of the cell buffer and the token pool via the sum of the two [3, 6]. This implies that by increasing the token pool capacity, the cell buffer can be eliminated without affecting the steady state throughput and blocking. This is desirable, if the network can handle larger bursts, since the delay due to a cell buffer can be reduced and the implementation cost of large token pool is smaller than the cost of a large cell buffer. It was shown that the departure rate of cells can be made independent of their arrival rate by increasing the size of the the token pool, δ, above some amount (approximately 10) [3]. The purpose of a cell buffer at this point is to handle a system in which token pool control can turn on only during the time of congestion scheme[3].

One disadvantage of leaky bucket scheme is that the bandwidth enforcement the token pool introduces is in effect even when the network load is light. In addition, leaky bucket is highly likely to mistake nonexcessive traffic as excessive [4]. In other words, cells will be lost even though long term average rate of the source is within the allocated bandwidth. To, solve this problem, a virtual leaky bucket was proposed [25, 2, 10, 11].

In virtual leaky bucket scheme, cell arriving at an empty token pool are marked and transmitted without a token, while those that have tokens are not marked. Marked cells are considered violators of allocated bandwidth since the call must have exceeded the allocated bit rate for some time for the token pool to be empty. Since bandwidth may still be available in the network, marking cells allows the call to exceed its allocated bit rate if it does not adversely affect other calls. If at some point along the path a marked cells reaches a congested link, it may be discarded so the throughput of the unmarked cells is not significantly affected. This not only allows us to take advantages of light network

load, but also allow a larger margin of error in determining the token pool parameters [4].

Some have argued that marking is ineffective and may degrade the performance of the unmarked cells [7]. However, buffer management schemes exist that give near optimal performance for unmarked cells [2]. Another short comings of marking is that it has no correlation with the user level data priority. The user has a better understanding of which cells are more important in delivery whereas the marking system determines priority regardless of user level priority.

To further decrease the effects of marked cells on unmarked cells, the use of a second token pool for marked cells was proposed [2]. Additionally, a spacer is used that imposes smoothing. When a cell is delivered into the network, its token is removed and the token enters a spacer. The tokens are discarded from the spacer at a rate β. The next cell at the head of the output queue cannot be transmitted even if tokens are available in the pool unless spacer is empty. This assure that the cells are transmitted at a rate less than or equal to β at all the times by inserting spaces between the cells. This effectively allows two levels of rate control whereas in the previously discussed schemes, there is only one level of control, i.e. marked cells can be transmitted at will.

2.3 Cell Prioritization

Due to the diverse characteristics of traffic streams competing for ATM resources, some form of prioritization must be in place to determine how cells of different classes of traffic are treated in the network. To provide multiple grades of service with ATM, we can use priority between and within service classes. Having determined priority of different services, we must handle the prioritized cells in an appropriate manner during cell discarding and scheduling. Priority in discarding scheme determines which cells are dropped when buffer overflow occurs. The order for cell transmission is determined in scheduling priority.

2.3.1 Priority Scheduling

A fixed, priority scheme is a simple scheme to serve multiple classes with various delay requirements. The traffic is classified into k fixed priorities. The input buffer is divided into k queues and arriving cells are placed in the corresponding queue. As long as the class 1 (highest priority) queue is not empty, cells in class 1 queue are served. When the class 1 queue becomes empty, class 2 cells can be served. When both class 1 and class 2 queue becomes empty, class 3 cells can be served and so forth. This method is disadvantageous for continuous bit rate traffic since it will always have priority. However, performance for lower classes is poor. The delay for lower classes may become intolerably large if there is large volume of high priority traffic.

A flexible priority scheme solves the problem of giving too much priority to one class. The basic idea is that cell in the lower priority also have some chance to transmit even if there are higher priority cells in the queue. Thus, lower priority cells that have waited for long time can preempt higher priority cells in the service order. Flexible priority disciplines have been shown to minimize weighted sum of the mean waiting times in an M/M/1 network [21].

2.3.2 Priority Discarding

Discarding of cells based on the priorities of the voice or video can be used to prevent and tolerate periods of congestion. Normally cells are accepted into the input buffer until the buffer becomes full. With multiple priorities the *push-out* scheme can be used to decide which cell can be dropped when buffer overflow occurs. When buffer becomes full, higher priority cell can *push out* lower priority cells. To maintain minimal throughput for lower priority traffic, push out can be limited such that high priority cell can push out cells of class j only if there are more than N_j cells of class j. The total number of cells lost is same whether the *push out* scheme is used or not, since push-out scheme determines which cells are to be discarded, not how many.

The push-out scheme gives nearly optimal performance but is difficult to implement. High priority cells must be able to discard a low priority cell at any location in the buffer.

This can be extremely difficult to implement. A practical alternative is to accept low priority cells in the buffer only if the total occupancy of the buffer is less than some threshold. This method has also been shown to give near optimal performance [2, 16]. Previously we discussed marking for purposes of bandwidth enforcement. A similar marking scheme can be used in voice or video coding with high priority (unmarked) cells corresponding to the MSP and low priority (unmarked) cells to the LSP. By combining bandwidth enforcement and coding priority, loss can be tolerated. If the bandwidth enforcement scheme will require some cells to be marked, the best way to determine which cells are marked is to mark those cells that would have least effect on service quality.

In addition to discarding cells when buffer overflow, preemptive discarding can be used in conjunction with other priority scheme, such as flexible priority, to reduce traffic. Preemptive discarding is based on knowing in advance that cells may eventualy be discarded. The dells that are likely to be discarded, particularly marked or low priority cells, can be discarded before more network resources are invested on them. Preemptive discarding not only relieves congestion at the node in which cells are discarded but alleviates other nodes of unnecessary traffic. There are four scheme that can be used to determine when when a cell might be preemptively discarded. A cell may be discarded if [22]:

- Upon arrival to queue, a load check indicates congestion at the node
- Time spent at the node exceeds a local deadline
- Time spent at the node exceeds a end-to-end deadline
- Time spent in the system plus the node exceed the end-to-end deadline.

Although a scheme using end-to-end deadlines is optimal, it is not practical. Because it discards cells if delivery to their destinations will be too late to be useful. Since synchronization of clocks in ATM is near impossible. It is not possible to determine *too late* cells. Hence, only the first and second methods above may be practical.

2.4 Reactive Controls

This type control takes action to recover from congestion once it has occurred.

2.4.1 Explicit Congestion Notification (ECN)

ECN is a mechanism by which the end system is kept informed about the congestion status in the network. In this protocol, each network element continuously sends its congestion informations to end points of the connections that are passing through it. This information is sent as single bit in the cell header. Each node detects congestion by monitoring the buffer occupancy and link utilization. Upon determining that congestion may occur in near future, a node sets an indication in the ECN field of all the passing cells whose connection are likely to be affected. Once the risk of congestion is abated, the congestion indicator is set in the cell header appropriately. In response to the network congestion indicator, the higher layer protocols are expected to start shaping the volume of traffic admitted to the network. This function may be achieved by implementing an adaptive window or adaptive rate procedure in the customer equipment.

The benefit of ECN is its ability to reduce the cell loss rate. As a result the retransmission of higher layer data units (such as frames) is greatly diminished, thus improving the effective throughput of the network during the congestion periods.

It is found that ECN is highly beneficial when the duration of congestion is at least an order of magnitude larger than the end-to-end delay. ECN is moderately useful when the congestion duration is of the same order as the end-to-end delay. The use of ECN should not be limited to data service. For example, variable bit rate codecs can change their coding scheme to produce a lower bit rate output when ECN is received.

2.4.2 Fast Reservation Protocol (FRP)

FRP uses in-band signaling to negotiate changes the information transfer rate of a connections. Two variations of FRP are being proposed: FRP/DT (Delayed Transmission) and FRP/IT (Immediate Transmission). According to FRP/DT, whenever there is a change

in transfer rate, at first it will send a special request cell, while continuing to transfer at the previous transfer rate. Upon receiving this request cell, network elements attempt to reserve bandwidth for this connection at the specified rate in the special cell. The newly specified rate may be less than the previous request, which indicates releasing of the resources. If this allocation is successful in all the nodes along the route, an acknowledgement is sent back to the source and source will be transmitting at the newly requested rate. The advantage of this protocol is reducing in signaling time and reliable transport of correlated cells belonging to a larger date unit. This protocol is suitable for those traffics which causes adverse impact on transmission in case of loss and can tolerate higher delay. If the end-to-end delay is very high and frequency of renegotiation is also high, using this protocol will result in inefficiency.

In FRP/IT information are sent without any renegotiation. If the node can accommodate the information then it transmit, otherwise it drops the whole burst of information. Every FRP/IT is burst followed by a resource release cell. FRP/IT is useful for delay sensitive traffic. The QoS guarantee for delay sensitive traffic may be affected in the presence FRP/DT traffic.

2.4.3 Backward Explicit Congestion Notification (BECN)

In BECN congestion information is returned direstly from the point of congestion back to the source for each virtual connection. The source adjust its cell transmission rate on each virtual connection. BECN capable of reacting faster than ECN. Since, the network itself generates the congestion feedback information, it is more robust against end systems that do not comply with the requirement of the scheme. The policing mechanism at the entrance to the network can itself perform the rate adjustment in response to the congestion feedback so that traffic from the non-compliant sources can be discarded. One disadvantages of BECN scheme is that network generates extra cells, which must be in acceptable range.

In this chapter, we have discussed various issues of congestion control in ATM networks. In Section 2.1, we described briefly some connection admission control schemes and their

advantages and disadvantage. Leaky bucket and its variations, as a policing and shaping function of congestion control is mentioned in section 2.2.1. Priority is required to facilitate multiple QoS, which is discussed in section 2.3. In Section 2.4, we briefly mentioned three reactive congestion control schemes.

CHAPTER 3

A Framework for Loss Sensitive Traffic in ATM

Retransmissions of packets that are dropped by the network is the primary cause of decreased network throughput. To increase network throughput every attempt should be made to reduce and perhaps eliminate it. Buffer overflow at the nodes cause cells to be dropped in the network. It is often seen that excepting the congested link buffer, other link buffers on the path of the virtual connections passing through the congested link are seldom used. In a properly designed network, the traffic load due to congestion should be distributed among all the buffers involving virtual connections. From these ideas we develop a congestion control scheme for loss sensitive traffic. In the following we describe the proposed scheme.

3.1 Hop by Hop Backward Explicit Congestion Notification (HHBECN)

We assume that there are separate buffers for each outgoing link of a switch and that the buffers for loss sensitive (like data) and delay sensitive (like video and voice) traffics are disjoint.

Let s be a switch in an ATM network and l one of its output links. Let $\mathcal{V}_l$ be the

set of virtual circuits using the link l. Consider a virtual circuit v that has switch s in its route. Let $p_v(s)$ be the predecessor switch of switch s for the virtual circuit v. Let $\mathcal{P}_l = \{p_v(s) | v \in \mathcal{V}_l\}$, i.e., $\mathcal{P}_l$ is the set of predecessor nodes of switch s for the virtual circuits using the link l.

In the following we assume that the output link l of switch s is congested. Congestion is indicated on a link, if the buffer occupancy of loss sensitive traffic in link buffer is more than a *congestion-threshold*, $T_c(l)$. When congestion is indicated, a *congestion-indicate*, CI, cell is sent to the switches that form the set $\mathcal{P}_l$. On receiving the choke cell members of $\mathcal{P}_l$ stop forwarding cells belonging to the virtual connections of $\mathcal{V}_l$ to the congested switch s and the CI is forwarded to its predecessor if the buffer occupancy of v is more then a threshold $TV_c(v)$. It should be emphasized that $T_c(l)$ is congestion-threshold for a link, whereas $TV_c(v)$ is for a virtual connection and their values are different. The value of $T_c(l)$ should be selected to ensure that enough buffer is left to store all the cells that may come before the preceding switches can act on the information in the CI cell. On the other hand, value of $TV_c(v)$ ensure that there is enough cells in the buffer to send continuously towards the congested switch, when congestion is abated.

The congested switch initiates a *congestion-relief*, CR, cell to the switches in $\mathcal{P}_l$ if buffer occupancy of the buffer for link l goes below a *congestion-relief* threshold, $T_r(l)$, for link l. On reception of the CR cell, the switches of $\mathcal{P}_l$ start sending cells for congested connections and forward the CR cell to those switches which were sent a CI cell previously. The value of $T_r(l)$ ensures that at a switch s cells in the out going link will not be emptied before any cell arrive from the previous switch, provided previous switch has at least a cell to send.

The suggested scheme is illustrated through the Fig. 1. In the figure S_1–S_3 are three sources, D_1–D_2 are two destinations, sw_1–sw_5 the intermediate switches and l_1–l_9 the links. Three virtual connections VC_1–VC_3 are as shown in the figure. When the occupancy of the buffer for output link l_4 of switch sw_4 increases beyond $T_c(l_4)$, congestion is indicated for the link. A CI cell is now sent to the members of $\mathcal{V}_{l_4}$, sw_2 and sw_4. On receiving the CI cell, sw_2 will stop forwarding cells belonging to VC_1 and sw_3 will stop forwarding cells of VC_2 to switch sw_4. VC_3 is unaffected. At the same time, the switch continues receiving

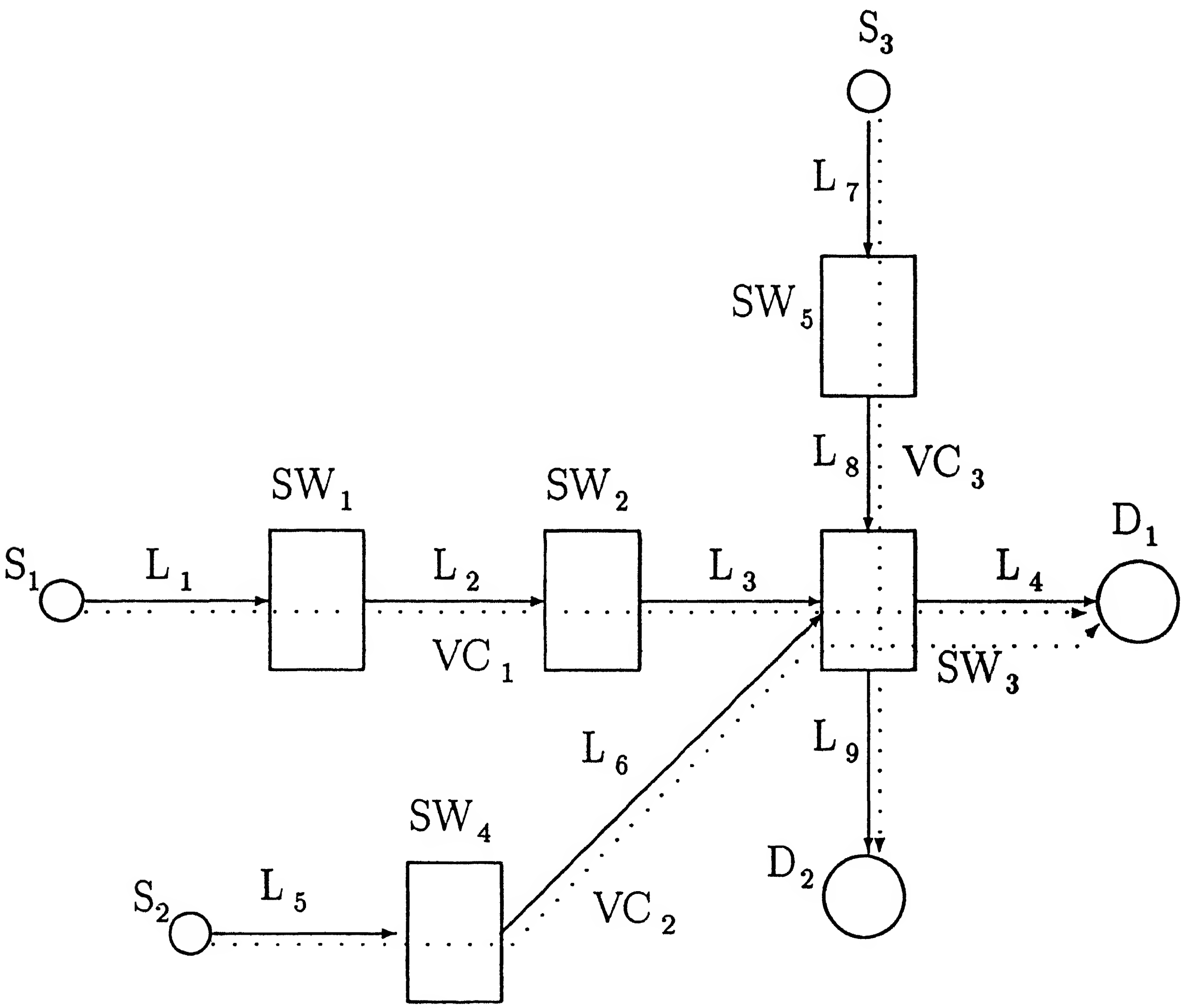

Figure 1: An Example

cells for choked virtual connections until buffer occupancy for that connection goes beyond a threshold $TV_c(v)$. If the buffer occupancy goes beyond $TV_c(v)$ for any choked virtual connection, the switch forwards the congestion information towards the source.

The congested switch sw_3 initiates a CR cell to the predecessors switches $\mathcal{P}(l4)$, if buffer occupancy for l_4 goes below congestion-relief threshold, $T_r(l4)$, for link l_4. On reception of a CR cell, sw_2 and sw_4 start forwarding cells of VC_1 and VC_2 respectively to sw_3 and forward the CR cell towards the source to those switches which were sent a CI cell previously.

Let the output link l of switch s be congested. Let B_l be the buffer size for loss sensitive traffic on the output link l. We derive the relation between $T_c(l)$, $T_r(l)$, $TV_c(v)$ and B_l, to provide high throughput and faster delivery of user information to the destinations without losing a cell due to congestion.

We select $T_c(l)$ such that all the cells that may come to s for link l before the switches in $\mathcal{P}$ receive the CI cell. For each loss sensitive virtual connection v passing through link l, let t_v be the propagation delay between s and predecessor switch of s for virtual connection v. Let p_v denote the peak rate for connection v. We obtain $T_c(l)$ as follows.

$$T_c(l) \leq B_l - 2\sum_{v=1}^{N} p_v t_v \tag{1}$$

where N is the number of connection through link l.

We select the value of $T_r(l)$ such that the link utilization of l is kept as high as possible. This is possible when s starts receiving cells from those sources which were choked before it empties the buffer of link l. That is,

$$T_r(l) \geq max(t_v)L \tag{2}$$

where L is the maximum transfer rate of the outgoing link.

In the event of a congestion, if the users are allowed to send their packets as close to the destination as possible, then the end-to-end delay of could be decreased. In other words, the packets should be stored as close to to the congested node as possible to allow transmission of data as soon as congestion abates. To allow for this, the switches should

have enough number of cells to transmit which can be continuously sent to the next switch when the congestion abates. Therefore,

$$TV_c(v) \geq 2t_v P_v \tag{3}$$

The following relations must hold to ensure zero cell loss and low end-to-end delay

$$T_c(l) > T_r(l) \tag{4}$$

$$T_c(l) \geq 2\sum_{v=1}^{N} < TV_c(v) \tag{5}$$

Condition 4 is obvious. The greater the difference between $T_c(l)$ and $T_r(l)$, the lesser would be the number of CI and CR cells generated by the switches. Condition 5 should be satisfied to reduce the false congestion indication. In other words, there should not be any congestion indication for all the virtual connections if some of them are choked by the successor switch of s (the switch immediately following s along the path towards the destination). Consider the following example. There are 3 connections (VC_1–VC_3) passing through a switch sw_1 and two of them (say VC_1 and VC_2) are choked by the successor switch sw_2 of sw_1. Then VC_1 and VC_2 are allowed to consume twice of their threshold $TV_c(1)$ and $TV_c(2)$ respectively. If the threshold of link l, is less than the twice the sum of $TV_c(i)$ of choked connections, then the third connection, VC_3, gets a false congestion notification. There is a chance of false congestion notification even though condition 5 is satisfied. This is because if some connections occupy more than $TV_c(v)$, before a congestion notification is sent to them. A possible solution for this problem is to restrict a virtual connections occupying more than $TV_c(v)$, by sending a CI cell to that virtual connection. This may increase the number of control cells generated by the network to intolerably high levels. An alternative solution is to increase the buffer size so that the threshold goes high, restricting a virtual connection occupying some multiple of $TV_c(v)$.

Let us consider an example to check the buffer requirements to implement HHBECN. For the sake of simplicity, let us assume that there is a switch at every 100 km. Hence, the propagation delay between two switches is 500 μsec. Suppose that the cumulative peak

rate of admitted loss sensitive traffic is ten times the link bandwidth, 155.442 Mbps. By Condition 5, the minimum threshold value to avoid false congestion notification is

$$T_c(l) = 2\sum_{v=1}^{N} TV_c(v)$$

From Equation 1, the minimum buffer requirement is

$$B = 6.0\sum_{v=1}^{N} P_v t_v$$

Therefore,

$$B = 6 * (10 * 155.442 * 1000000) * (500 * 1/(1000000))/8 = 582.894KB$$

If we can have bigger buffer size, the congestion notification can be sent on virtual connection level rather than in link level.

In the following section, we propose a congestion control mechanism for all type of services provided by B-ISDN using HHBECN. It should noted that, to use use HHBECN effectively, the following mechanism or a similar congestion control must be used.

3.2 The Congestion Control Mechanism

Three classes of services are considered here. The QoS of these are as follows.

- **Class 1** : No cell loss and no cell delay. This service class cannot be considered for statistical multiplexing and hence, must be given the highest priority.
- **Class 2** : Bounded delay and very stringent cell loss of the order of 10^{-10}. Due to its variable bit rate, the statistical multiplexing gain can be achieved, but the bounded delay should be taken into consideration.
- **Class 3** : No cell loss and no stringent delay requirement. High statistical gain can be achieved, but care must be taken to achieve no cell loss to increase the throughput.

3.2.1 Admission Control

A new call is *rejected* if cumulative average bandwidth requirement of all the existing connections and the new connection is more than the link bandwidth. The admission control allocates the peak bandwidth to Class 1. When a Class 1 call arrives, the QoS requirements of Class 2 traffic are computed as the difference of link bandwidth and cumulative peak bandwidth of Class 1 traffic. This computation also takes the requirement of the new Class 1 call into consideration. The new Class 1 call is accepted only if the newly calculated QoS requirement does not violate QoS agreement of the existing Class 2 traffic. Similarly, a new Class 2 call is accepted only if admission of the new call does not degrade the QoS requirement of existing connections. The calculation for Class 2 traffic is carried out as the difference of link bandwidth and cumulative peak bandwidth of Class 1 traffic.

The bounded delay requirement of Class 2 traffic can be guaranteed by providing a small buffer. For example, if the bandwidth available for Class 2 traffic is 100 Mb and the maximum jitter allowed is 5 msec then a buffer size of (100mb*5msec/(53*8)) = 1179 cells will guarantee an upper bound of 5 msec on the delay. Now, the loss probability for Class 2 connections is the probability that buffer overflow will occur.

A Class 3, loss sensitive, call is accepted if Conditions 4 and 5 are satisfied. These conditions guarantee no cell loss and reduce the probability of false alarm of congestion control for the loss sensitive traffic.

3.2.2 Usage Parameter Control

The peak rate for Class 1 traffic is monitored at the network entry point using a simple leaky bucket scheme. The leaky bucket scheme that we follow works as follows. The tokens are generated at the peak rate, having a token buffer size of one. A little tolerance in the peak rate is permissible. For Class 2, the traffic is monitored at the network entry point to comply with the peak rate, average rate and burst size. This class of traffic is monitored using virtual leaky bucket with spacer [2]. Class 3 traffic goes under traffic shaping in the internal switches, so that they are not transmitted more than the peak rate.

3.2.3 Priority Scheduling

We use a fixed priority scheme. The different classes of traffic are kept in separate queues. Class 1 traffic is transmitted first. If Class 1 queue is empty then Class 2 traffic is sent. Class 3 traffic is given the lowest priority and it is sent only when there is no cells in the Class 1 and Class 2 queue. In this way, Class 1 traffic will be delayed by at most the number of Class 1 connections admitted multiplied by the time required to send a cell at the link speed.

3.2.4 The Reactive Control Mechanism

HHBECN is the reactive control for this mechanism. HHBECN control is only for loss sensitive traffic. Every virtual connection is associated with a counter to keep track of the number of cells in the buffer. A counter is maintained for each link to record the total number of loss sensitive cells in the buffer. Every time a cell is received or transmitted, the link counter and corresponding virtual connection counter is incremented or decremented, respectively. When the link counter goes beyond the threshold of the link, a CI cell is sent to all the predecessor switches, for the connections going through the congested link. On receiving the CI cell for a connection, a switch stops sending cells for that connection. The switch continues receiving cells for the same connection until corresponding virtual connection counter goes beyond the particular virtual connection threshold. If virtual connection counter goes beyond the particular virtual connection threshold a CI cell is sent towards the source. If the source receives the CI cell it stops the transmission of cells until it received a CR cell.

Congested switches transmit a CR cell to the predecessor switches of the virtual connections through congested link, if buffer occupancy goes below a congestion-indication threshold. On receiving a relieving cell, the choked connections start sending cells, and forward the CR cell to those switches which were sent a CI cell previously.

3.3 Summary

In this chapter we proposed a complete congestion control scheme using HHBECN. In Section 3.1, a detail description of the proposed scheme HHBECN for loss sensitive traffic given. In Section 3.2 a congestion control mechanism, including how admission of connections have to be control, what parameters of different type of services must be monitored, how to assign priorities, and also reactive control during the congestion, is presented. In the next chapter we discuss the implementations, and simulations and results in the chapter 6 obtained for the above mentioned framework.

3.3 Summary

In this chapter we proposed a complete congestion control scheme using HHBECN. In Section 3.1, a detail description of the proposed scheme HHBECN for loss sensitive traffic given. In Section 3.2 a congestion control mechanism, including how admission of connections have to be control, what parameters of different type of services must be monitored, how to assign priorities, and also reactive control during the congestion, is presented. In the next chapter we discuss the implementations, and simulations and results in the chapter 6 obtained for the above mentioned framework.

Chapter 4

Implementation

To setup an ATM network, which involves fibre optic cables and switches, is costly. Before we set up such a network we must check whether it is feasible, and cost-effective in variety of environments. To test our proposed scheme, we have used an ATM simulator AtmSim [23], which is capable of emulating a virtual reality model for ATM.

4.1 About AtmSim

AtmSim is a simulator for studying the dynamic behavior of congestion control or resource management schemes in ATM network. It provides user with a way of specifying such networks to observe their behaviors. The input to the simulator is a description of network topology, specification of sources, route of the virtual connections, and output files names. The output statistic produced by AtmSim includes - number of cells pass through each link, maximum queuing delay experienced in a switch, average euing delay experienced in a switch and other similar informations. The input to the simulator easy to specify and similar to the NetLanguage used by the simulator REAL.

AtmSim is a discrete event simulator, having the capability of switching cells from one switch to another. AtmSim has the capability of simulating arbitrarily large ATM network, with any kind of topology. AtmSim runs on a single processor rather than in distributed processors. In a discrete event simulator, running in one process has a better control over time than an distributed one. For example, in a distributed simulator different

processes will run in different processors to exploit the parallelism of the network being simulated. In such situations, one cannot impose an order on the events to happen. On top of that in unix like environment a process may be delayed due to load on the system. As a result, it is not possible to obtain accurate delay characteristics, cell loss and etc.

In AtmSim, various events of ATM communication are discretized and executed one at time. The events are kept in an eventqueue in the increasing order of time of occurring. Each event is associated with a time-stamp, event type, a procedure name and a set of parameters.

The procedures are assigned according to the event type. For example, SendCells() is the procedure for the event type sendCell. The procedure is executed at the time specified in the time-stamp of the event. The parameters in the event determines where the event will occur. For example, sendCell event has two parameters switch number and link number which implies that, at the given link number of the specified switch, some cells has to be transmitted at the time specified in the time-stamp.

There are five type of events in AtmSim, *viz.* genCell, sendCell, and three events for output - switchOpEvent (outputs for switch), linkOpEvent (outputs for link), and vcOpEvent (outputs for a virtual circuit). The generic AtmSim assumes that connection are setup before it executed. According to the number of connections passing through a link, the look up table for vpi vci and link is constructed. The simulator assigns an unique triple of vpi, vci, and link number to each connection going through a particular switch.

In the **genCell** event a number of cells are generated and appended to appropriate buffer. The cells are generated for the source specified in the parameter list. This event trigger sendCell event for the cells that were generated. This event invoke another genCell event to be executed for the same source, specified in the parameter list, to continue on generating cells as long as the virtual connection is alive.

In the execution of **sendCell** event, a number of cells, computed from the various condition of the network is transmitted from one switch, specified in parameter list, to next switch in the given link. In this event all the output data structure updated. This event may result in invoking other sendCell event in the next switch for the new cells being

transmitted.

switchOpEvent, **linkOpEvent**, and **vcOpEvent** writes to the output files, specified in the input file for the simulator, about output information in the output data structure for switch, link, and virtual connection respectively. User can specify separate file for each switch or for each link or for each virtual circuit.

4.2 Modification and Addition to AtmSim

To simulate the scheme mentioned in the previous chapter 3.2 we have added some features to AtmSim and modified AtmSim to incorporate these features. ATM admits new connection depending on the resource available in the network. The availability of the resource is determined by Connection Admission Control (CAC) algorithm. A connection setup module is added to AtmSim. As a result, four new types of events are identified. A brief description of each of them is given below.

connSetup The purpose of this type of event is to carry the request of the source, for a new connection towards the destination. At the execution of this event, it calls the procedure ConnSetup(). All the new events use the same parameters *viz.* `switchNumber` and `sourceNumber`. The procedure ConnSetup() checks for the availability of the resources and QoS requirements of the existing connection, depending on the connection admission control schemes being used, and decide whether to accept the new connection. If the connection is accepted, the switch inserts an entry in the look up table and buffer queue for the corresponding connection. Also it triggers another connSetup event for the next switch along route of the connection. If it reaches the destination switch, invoke a connAccept(described below) event conveying that the connection is accepted by all the network element along the path. A connReject event (mentioned below) is invoked towards the source, in case, required resources are not available.

connAccept As mentioned above this event is started by the destination to indicate that all the network element along the route decided to accept the connection. A

connAccept event invoke another connAccept event until it reaches the source. This event travel exactly in the opposite direction of the connSetup event. At the source, it initiate a genCell event for start transmitting cells towards destination.

connReject This event signifies that there is not enough resources to admit the connection asked for. This event travels from the congested node towards the source. On top of the passing the connection reject information towards the source by the intermediate nodes, the intermediate node remove the entry in lookup table and buffer queue that was allocated for that connection. At source it try to establish again after a retry duration specified in the input file for a maximum number of times also taken from input file.

connRelease This event is invoked by source once the time to live for the connection expires. This results in releasing of resources in all the network elements, the network elements also remove the entry from the look up table and buffer queue.

In Hop by Hop BECN scheme, during the congestion, it sends back a choking cell towards the source, and a relieving cell is sent when congestion abates. To implement this scheme two events are added to the list of types of event. They are **choking** and **relieving**.

choking When the buffer occupancy goes over the threshold the switch initiate a choking event towards the sources, to all the connections, passing through the congested link. A choking event invoke the Choking() procedure, which first check the buffer occupancy of the corresponding connection. If buffer occupancy is more than the threshold, as mentioned in the Section 3.1, another choking event is triggered towards the source for that connection. The transmission of cell is stopped, for the virtual connection as soon as a choking event is received by the switch. When a source receives a choking event it continue on generating cell until the window of the source exhausted.

relieving A relieving event is sent to all the connections going through the congested link, when buffer occupancy is less than the congestion abating mark. relieving

event calls the procedure Relieving() and it initiate a sendCell event to start sending again towards the destination.

4.3 Source Modeling

The accuracy of observation of any congestion control scheme highly depends on the accurate modeling of sources. One way is to observe the source behavior of a real network on which the scheme is intended to test, and use the collected data for testing the performance of the intended scheme. In this way, reaction of the sources to scheme cannot be realistically modeled, but this gives a nearly accurate behavior of the sources. Another way is to use mathematical model to emulate behavior of the sources. There has been a lot research over the years to model different type of sources, they are good enough for studying the behavior of a congestion control scheme. For implementation of HHBECN, we have gone for the later option, for practical difficulty of getting the first one.

We have modeled three different types of source *viz.* data source, packetize voice source, and video source. The mathematical model used for these source are described below.

4.3.1 Voice source

An arrival process of cells from a voice source if fairly complex due to the strong correlation among the arrivals. The arrival process of a voice call and their durations can be characterized by a Poisson process and exponential distribution, respectively. Within call talk-spurts and silent periods alternate. During the talk spurt voice cells are generated periodically at the rate of 64kbps. The correlated generation of voice cells within a call can be modeled by an Interrupted Poisson Process(IPP) [15, 19, 17, 8, 13]. In an IPP model , each voice source is characterized by ON (corresponding to talk-spurt) and OFF (corresponding to silence duration) periods, which appears in turn. The transition from ON to OFF occurs with probability β, and transition from OFF to ON occurs with probability α. In a discrete time case, ON and OFF periods are geometrically distributed with mean $1/\beta$ and $1/\alpha$, respectively. Cells are generated during ON periods according to a

Bernoulli distribution with rate λ; no cells are generated during OFF period.

4.3.2 Video Source

Digital video is expected to become a major traffic component in B-ISDN's. Application such as video conferencing, and switched TV, imposes a very large bandwidth requirements on network. We assume video source generating 25 frames/s. Each frame consists of approximately 270, 000 pixels. A variable bit rate video coding scheme is used to encode a frame. A *Continuous-State Autoregressive Markov Model* [18]for modeling the mentioned coding scheme for video traffic. We model the coder rate as a continuous-state, discrete-time stochastic process. Let $\lambda(n)$ represent the bit rate of a single source during the nth frame. A first-order autoregressive Markov process $\lambda(n)$ is generateed by the recursive relation

$$\lambda(n) = a\lambda(n-1) + bw(n) \tag{6}$$

where $w(n)$ is a sequence of independent Gaussian random variables and a and b are constant. Assume that $w(n)$ has mean η and variance 1. Further assume that $|a| < 1$; thus, the process achieves steady state with large n. The steady state average $E(\lambda)$ and discrete autocovarience $C(n)$ are given by [18]

$$E(\lambda) = \frac{b}{1-a}\eta \tag{7}$$

$$C(n) = \frac{b^2}{1-a^2}a^n \qquad n \geq 0 \tag{8}$$

The autocovarience is exponential and can fit the experimental data. The steady-state distribution of λ is Gaussion with mean $E(\lambda)$ and variance $C(0)$. From the measured data it was found that :

$$E(\lambda) = 0.52 bits/pixel$$

$$C(n) \simeq 0.0536 \times (e^{(-0.13)^n} (bits/pixel)^2$$

The discrete autocovarience $C(n)$ is obtained from the experimental fit $C(\tau) = 0.0536$ $\times e^{(-3.9\tau)}$ by sampling at $n/\tau = 30$ frames/s. Matching 7 and 8 with the measured data,

it was found [18],

$$a \simeq 0.8781 \qquad b \simeq 0.1108 \qquad \eta \simeq o.5772$$

Continuous-state autoregressive model provides a rather accurate approximation of the bit rate [18]. A better matching may be achieved if its order increases to include the influence on $\lambda(n)$ of several past values $\lambda(n-k), k \geq 1$. In this case, the autocovariance will be the sum of the several exponentials. Nevertheless, since a single exponential fit was found to dominate the decay of the autocovariance.

To generate gaussian noise the following algorithm is used.

Algorithm for Generating Gaussian Noise

```
begin
    repeat
        generate two uniform random number u1 and u2
        x = -ln(u1)
    until (u2 ≤ exp(-(x - 1)/2))
    generate a random number u3
    if (u3 ≤ 0.5)
        gauss = η + σx
    else
        gauss = η - σx
end
```

Where η denotes mean and σ denotes standard deviation.

In the implementation, for each frame, number of bits per pixel is calculated using equation 6, then multipling with number of pixels per frame we get the number of bits in that frame. Then computing number of cells per frame, cells are generated in equal interval in that frame.

Assuming that mean bits per pixel is 0.52, the average rate of a video connection is $= 270000 \times 25 \times 0.52 \times 53/44 = 4.228$Mbps. where each cell contains 44 bytes video code.

4.3.3 Data Source

Three parameters are used to model the data source, *viz.* average rate, peak rate and burst size. Cells from data source arrives as burst, at peak rate. In the model we used, cells in a burst is geometrically distributed and inter burst gap are exponentially distributed. Let P_i, A_i, and B_i be the peak rate, average rate, and mean burst size of a virtual connection i. The burst size is given in number of cells. The number of cells that are generated in a burst by tossing a coin, with probability of success in each toss is $B_i/(B_i + 1)$. The mean of inter cell gap is $((P_i A_i)/(B_i \times 53 \times 8)(P_i A_i))$.

CHAPTER 5

Simulations and Results

We have chosen simulation as the mechanism for for evaluating the performance of congestion control scheme. We have selected a simple topology, and simulate with a variety of traffic characteristics.

For simulation, voice source was not considered for its insignificant cell generation rate. In all the simulations, link bandwidth is assumed to be 155.442 mbps. Simulation were carried out for 1000 msec. We have not used any connection admission algorithm which can calculate the loss probability of delay sensitive traffic for a bounded delay requirement. We use simulation to decide how many video connections (delay sensitive) can safely be multiplexed without any loss. In all the simulation, the "maximum" delay allowed is 2 msec for delay sensitive traffic. We achieve 2 msec delay bound by providing a small buffer, as mentioned in Section 3.2. A simple network as shown in the Figure 2, is used for finding out the number of video connections that can go through a link. Here, $S_1, S_2, \ldots, S_n$ are the sources and D is the destination and C_1 and C_2 are the switches where congestion may occur. We have monitored the performance parameters for the link $C_1 - C_2$. The characteristics of video traffic is as described in the Section 4.3.2. The average data rate for each connection is 4.228 mbps. Video connection with these characteristics is used for all other simulation. It was found that 22 video connections can be safely multiplexed, without any cell loss.

Using the same network model we have simulated for loss sensitive traffic and checked the performance at the link $C_1 - C_2$. We accepted connection as long as cumulative

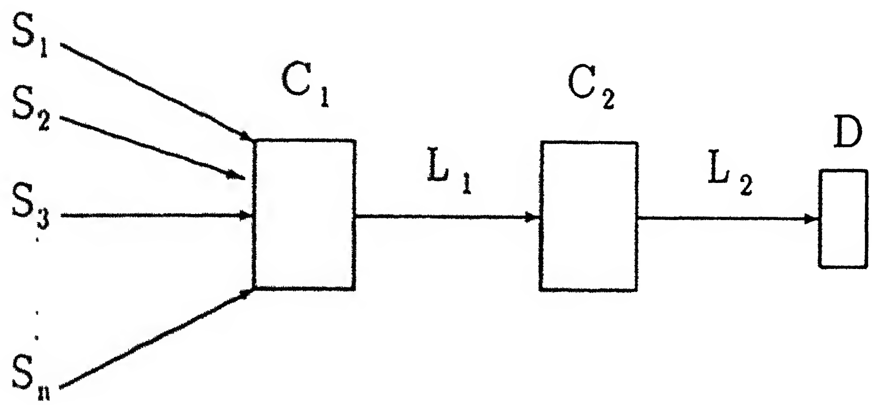

Figure 2: Topology 1

simulation number	peak rate Mbps	average rate Mbps	burst size cells	cells sent	average delay msec	chok/ relieve graph	link utilization graph
1	50	5	200	345812	7.5251	3	4
2	50	5	500	346744	9.2058	5	6
3	100	10	200	343509	7.5925	7	8
4	100	10	500	340166	9.8531	9	10
5[1]	100	10	500	112253	11.1346	11	12
6[2]	100	10	500	265659	9.6045	13	14

Table 1: Simulations results for loss sensitive traffic of the link L_1

average bandwidth is less than the 98 percent of the link bandwidth. Four simulations were performed to observe the effect of the loss sensitive connections among themselves. All the four simulations are carried out using a homogeneous characteristics of the connections. The ratio of the peak to average of the connections are 10. Two burst sizes are used 200 cells and 500 cells.

Using both loss sensitive and delay sensitive traffic we did two simulations. In this case for simplicity of analysis we take loss sensitive connection of the same characteristics. The peak rate of the loss sensitive source is 100 Mbps and average rate is 10 Mbps. First simulation is for 22 delay sensitive connections and rest is loss sensitive connection and second one is 7 delay sensitive connections and rest are loss sensitive connections.

There are two types of graphs generated from the simulations, one is for link utilization and the other is for number of choking/relieving cells generated. For link utilization graphs, on x-axis time is plotted in msec, and on y-axis percentage of link utilization is plotted in the scale of 100:1. For the choking/relieving graph x-axis is time is plotted in msec and y-axis is total number of CI and CR cells sent to the all the predecessor of the link $C_1 - C_2$ is ploted.

Comparing the simulation 1 and simulation 2, we observe that the increase in the burst size increases the average delay. From simulation 1, 2, 3, and 4, it is apparant that there is no effect of peak rate on average delay as long as cumulative peak and average rates are same. This is because the link utilization are almost 100 percent, as we observe from the graph. Hence, HHBECN is successful in increasing the link utilization about 100 percent.

Number of CI and CR increases linearly with the increase of the peak rate and burst size, as it is apparent from the Figure 3, 5, 7, 9, 11, 13. In the Figure 9, we observe that about 1100 CI and CR cells sent by the switch, which is only 0.65 percent of the total cell sent in link.

Looking at the simulation 5 and 6 we found that, even in the presence of delay sensitive traffic, loss sensitive traffic can increase the link utilization to 100 percent using the HH-BECN scheme. Number of CI or CR generated is less than the other simulation because the amount of loss sensitive traffic present in the lesser than the previous simulations.

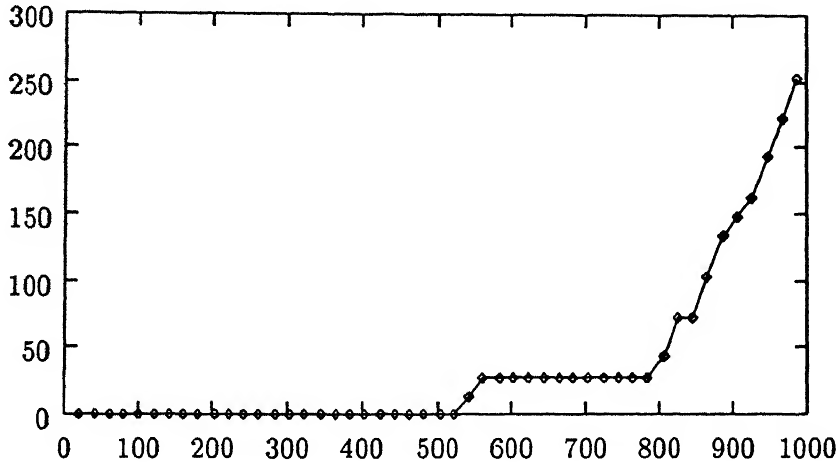

Figure 3: Choking and Relieving cells generated for simulation 1

5.1 Summary

We made the following observation about HHBECN:

- Link utilization can go up to 100 percent.
- Number of CI and CR cells generated by network increases linearly with the increase of burst size and peak rate.
- Average delay of the loss sensitive traffic is not affected by peak rate.

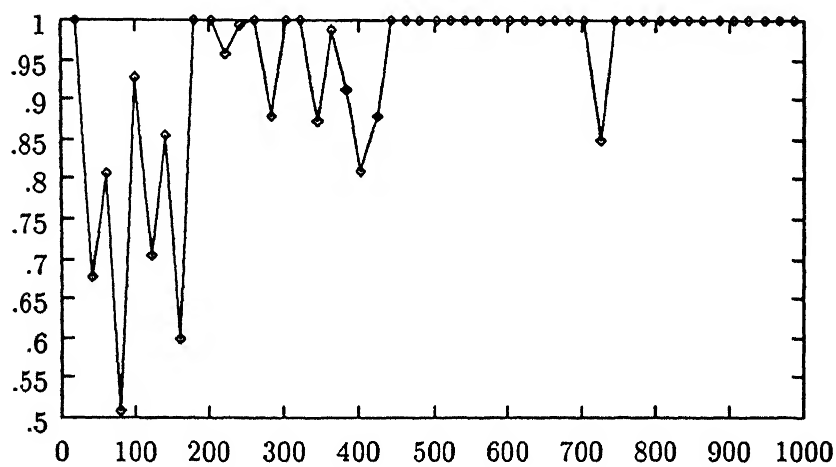

Figure 4: Link utilization for simulation 1

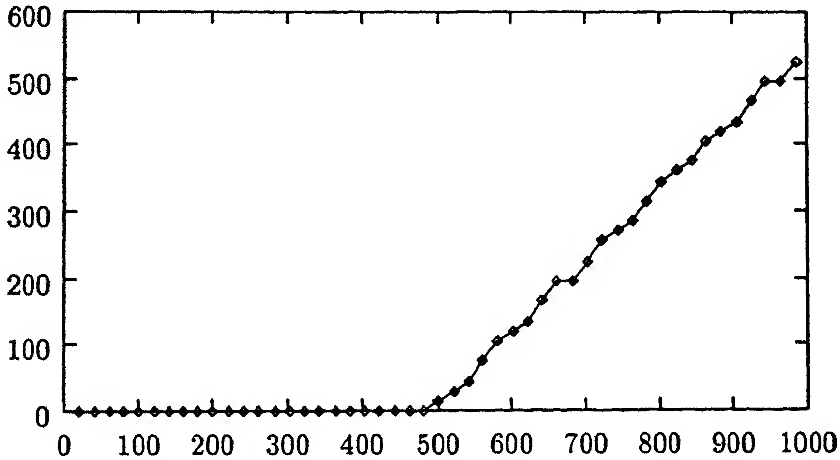

Figure 5: Choking and Relieving cells generated for simulation 2

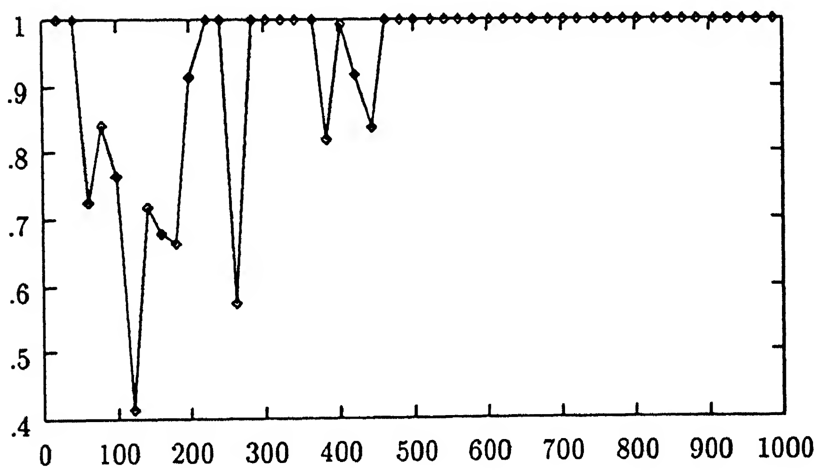

Figure 6: Link utilization for simulation 2

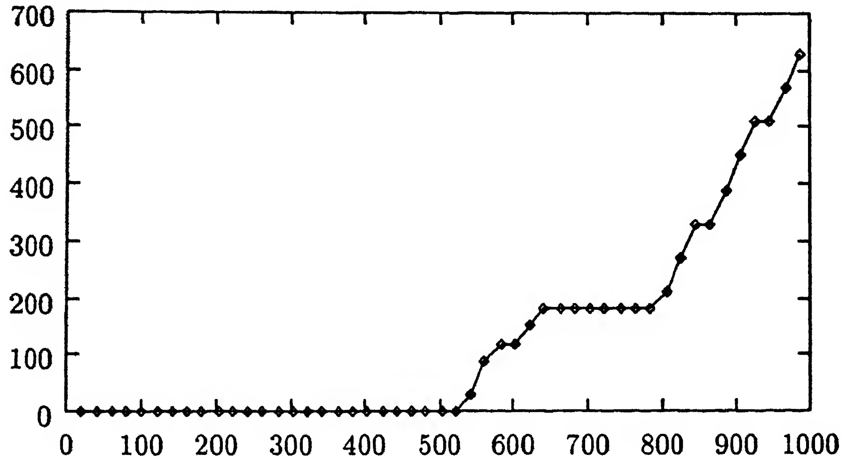

Figure 7: Choking and Relieving cells generated for simulation 3

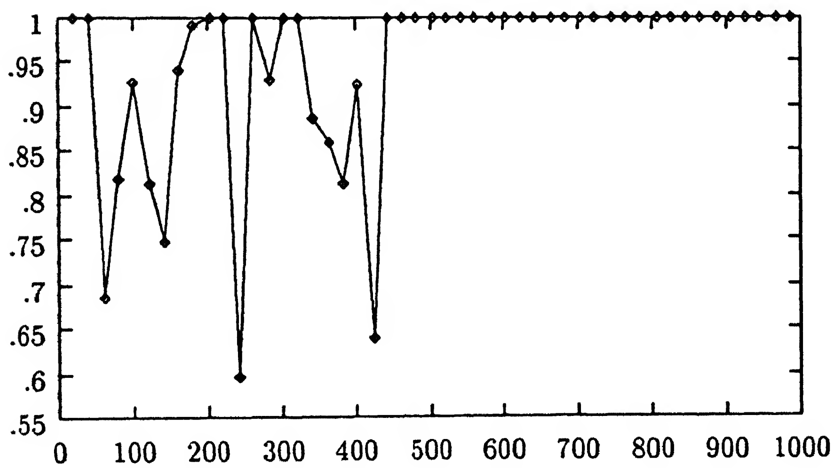

Figure 8: Link utilization for simulation 3

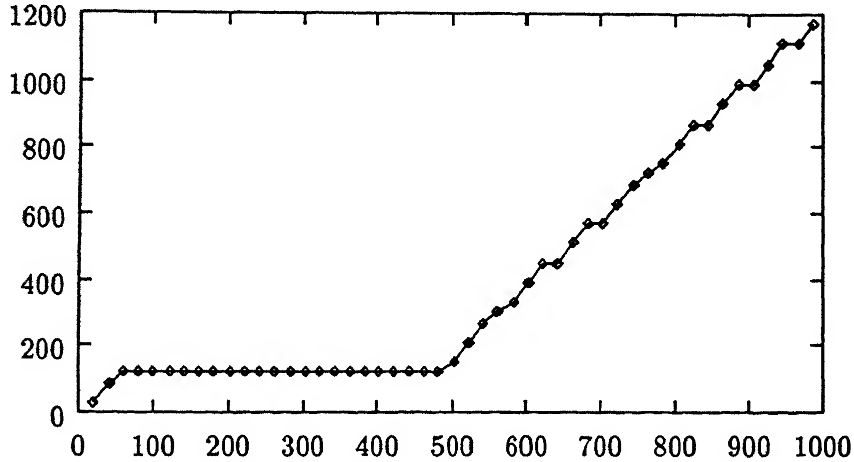

Figure 9: Choking and Relieving cells generated for simulation 4

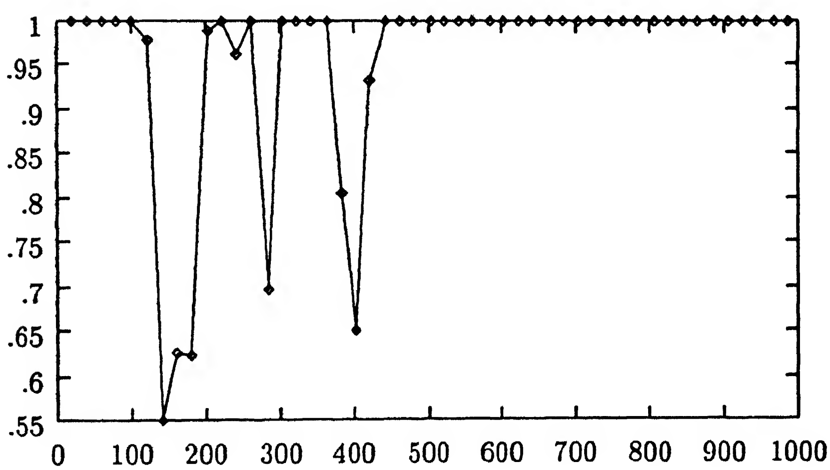

Figure 10: Link utilization for simulation 4

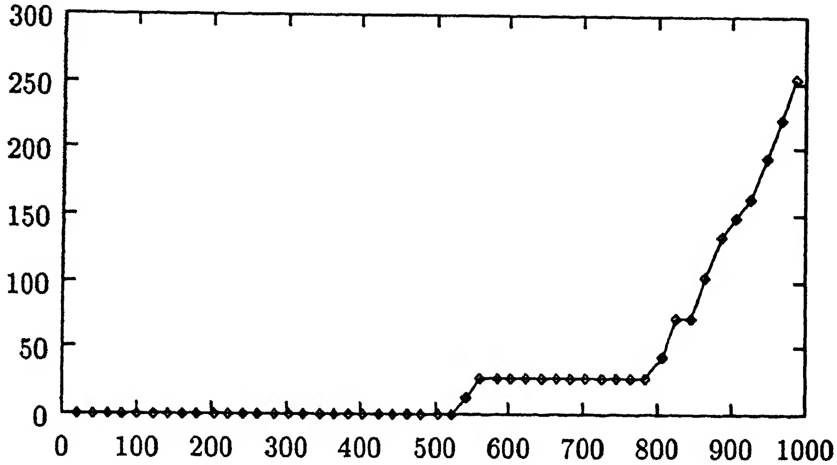

Figure 11: Choking and Relieving cells generated for simulation 5

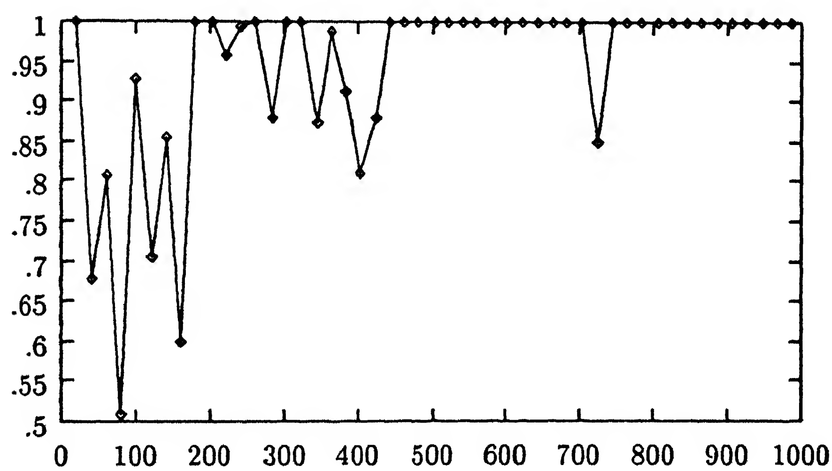

Figure 12: Link utilization for simulation 5

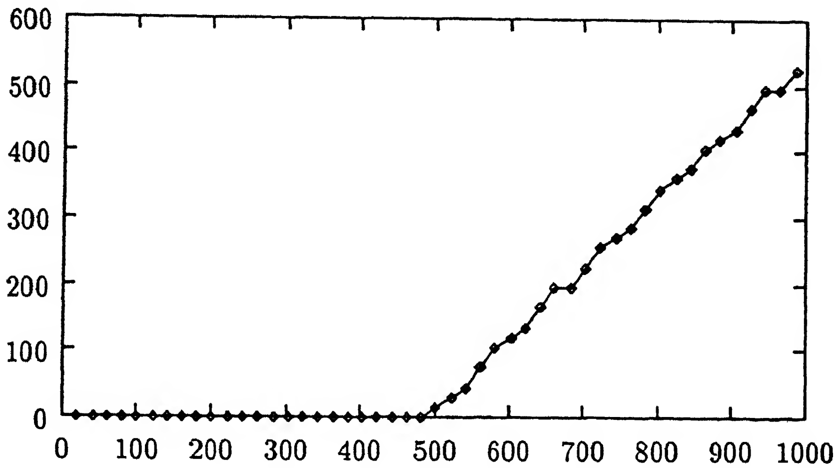

Figure 13: Choking and Relieving cells generated for simulation 6

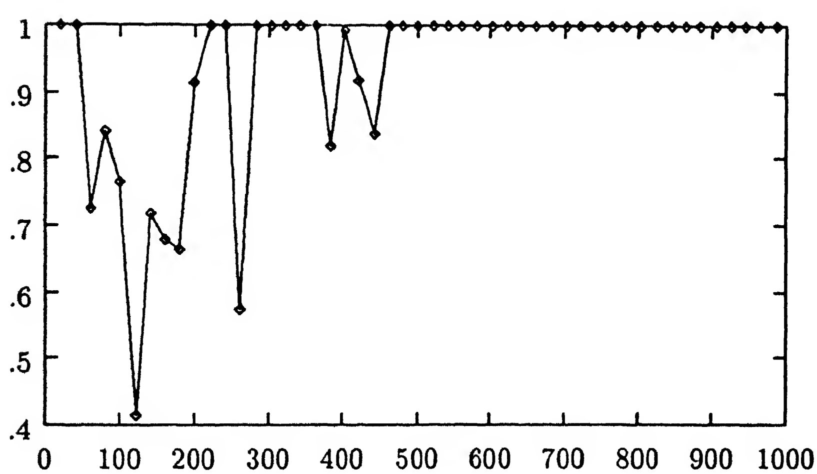

Figure 14: Link utilization for simulation 6

CHAPTER 6

Conclusion

ATM is a new and emerging technology. It can prove to be a better alternative than the existing communication media only if the statistical gain of ATM is significantly higher compared to that of other media. It should be noted that other media have an edge over ATM, because they are cheap, easily available, and incur less overheads compared to 9.5% overhead of ATM in the form of cell header. The statistical gain of ATM directly depends on the *effectiveness* of congestion control. Congestion control in ATM is, obviously, of critical importance and demands thorough investigation.

The following sections discuss the state of the art for each of the steps involved in congestion control in ATM. At the end, we list the advantages and shortcomings of HHBECN and suggest some possible extension of the work reported in this thesis.

6.1 Admission Control

Admission control is responsible for limiting the traffic flow, and guaranteeing the required QoS. It is is an important and complex step in congestion control. Any admission control scheme has to be real time, simple to implement and robust. Of course, admission control scheme makes the decision depending on the other steps in congestion control. As of now, the ATM traffic characteristics are not very well understood. An early deployment of ATM is essential [12] to analyze its traffic characteristics. A better understanding of ATM traffic characteristics would involve the following steps: observe the behavior of the

incoming traffics; develop models based on the observations; develop resource allocation rules from the models and continue observation and modify resource allocation rules, as required. Note that the process is dependent on a good model of incoming traffic but does not directly force the traffic to be "well behaved".

6.2 Usage Parameter Control

UPC is essential to ensure that users are behaving well. It should be simple and should as fast as time requires to transmit a cell.

6.3 Reactive Control

Reactive controls are necessary to give feedback to end users, so that user can take necessary steps to cooperate with the network. For example loss sensitive source can reduce the cell generation rate, video or voice source set the CLP bits for indicating that unimportant cells, or they can change their coding scheme to produce a lower bit rate output. More research is required in reactive control which will possibly increase the utilization of the network.

6.4 HHBECN and Further Extension

In this thesis presented the HHBECN scheme, which guarantee no loss for the loss sensitive traffic. To our knowledge this is the first step towards to achieve better utilization of the network resources, by eliminating retransmission. All of the proposed reactive controls in in Section 2.4 do not perform well in case of transient congestion. In thesis presented the HHBECN, is capable of controlling both transient congestion as well as long term congestion. In case transient congestion the congestion may abate before the congestion indication, sent by congested switch towards source, reach source. In this case, intermediate nodes takes care of the congestion. Since HHBECN work hop by hop, it overcome the problem of large delay bandwidth product present in ATM networks. Since, the network

itself generates the congestion indication cells, it is more robust against end systems that do not comply with the requirement of the scheme.

On the other hand, HHBECN increases the hardware complexity of the switch and also increases memory requirement. Another disadvantages of this scheme is that it generates extra control cells. But from the simulation results it was seen that generation of extra cells is less than 1 percent of the cell being transmitted.

The proposed scheme should evaluated in more number of network topology with their varying load. Another possible to extension of this work is compare the performances of the scheme with other schemes, in IP kind of source in which a packet is retransmitted if a cell loss occur.

Bibliography

[1] A Alles. Tutorial: ATM in private networking. Hughes Lan System, 1993.

[2] K Bala, I Cidon, and K Shoraby. Congestion control for highspeed packet switch network. In *IEEE INFOCOM*, 1990.

[3] A W Berger. Performance evaluation of a rate control throttle where tokens and jobs queue. In *IEEE INFOCOMM*, 1990.

[4] A W Berger, A E Eckberg, T C Hou, and D M Lucantoni. Performance characterizations of traffic monitoring and associated control mechanisms for broadband packet networks. In *IEEE INFOCOMM*, 1990.

[5] J-Y Le Boudec. The Asynchronous Transfer Mode: A tutorial. *Computer Networks and ISDN Systems*, 24:279–309, 1992.

[6] M C Chuah and R L Cruz. Approximate analysis of average performance of (s, r) regulators. In *IEEE INFOCOMM*, 1990.

[7] C A Cooper and K I Park. Toward a broadband congestion control stretegy. *IEEE Network Mazagine*, 1990.

[8] J N Daigle and J D Langford. Models for analysis of packet voice communication system. *IEEE J Selected Areas in Communication*, 1986.

[9] Prycker M de. *Asynchronous Transfer Mode Solution for Broadband-ISDN.* Ellis Horwood, 1993.

[10] A E Eckberg, D T Luan, and D M Lucantoni. Bandwidth management: A congestion control strategy for broadband packet networks characterizing throughput-burstiness filter. In *Int'l Teletraffic Congress Speciliast Seminar, Adeldia, Australia*, 1989.

[11] G Gallassi, G Rigolio, and L Fratta. Atm: Bandwidth assignment and bandwidth enforcement policies. In *IEEE GLOBECOM*, 1989.

[12] H Gilbert, O Aboul-Magd, and V Phung. Developing a cohesive traffic management strategy for ATM networks. *IEEE Communications Magazine*, 1991.

[13] I Ide. Super position of interrupted poision processes and its application in packetized voice multiplexers. In *12th Int. teletraffic congress, Torino, Italy*, 1988.

[14] M Iiyas and H T Mouftah. Performance evaluation of congestion avoidance in Broadband-ISDNs. In *IEEE ICC*, 1990.

[15] T KamiTake and T Suda. Evolution if an admission control scheme for an ATM network considering fluctuation in cell loss rate. In *IEEE GLOBECOM*, 1989.

[16] H Kroner, G Hebturne, and A Grevay. Priority management in ATM switching nodes. *IEEE J of Sel Areas in Comunication*, 1991.

[17] S-Q Li. Study of information loss in packet voice system. *IEEE Trans Commun*, 1989.

[18] B Maglaris, D Anastassiou, P Sen, Karlsson G, and Robbins J D. Performance model of statistical multiplexing in packet video communication. *IEEE Transactions on Communications*, 36(7):834–844, July 1988.

[19] M Murata, Y Oie, T Suda, and H Miyahara. Analysis of a discrete-time single server queue with bursty inputs for traffic control in ATM network. In *IEEE GLOBECOM*, 1989.

[20] S R Li. Algorithms for flow control and call setup in multihop Broadband-ISDN. In *IEEE INFOCOM*, 1990.

[21] H Satio. Optimal queuing discipline for real time traffic at all switching nodes. *IEEE Trans. Communication*, 1990.

[22] H Schulzrinne, J F Kurose, and D Towsely. Congestion control for real time traffic on high speed networks. In *IEEE INFOCOMM*, 1990.

[23] S B Vinod. Design and implementation of a ATM simulator. Master's thesis, I I T, Kanpur, in praparation.

[24] W Wang and T N et al Saadawi. Bandwidth allocation for ATM networks. In *IEEE ICC*, 1990.

[25] G M Woodruf and R Kostipaiboon. Multimedia traffic management principles for guaranteed ATM network performance. *IEEE J Sel Arears Commun*, 1990.

[26] S Yazid and H T Mouftah. Congestion control methods for BISDN. *IEEE Communication Magazine*, 1992.